Causal effect identification and estimation

Causal Inference Assignment 2

Anna Pauxberger

04 November 2019

Introduction

The following paper looks at methods to estimate the effect a Democratic Senate majority has on drug approval time in the United States. The dataset used originally was created by Carpenter (2002) to look at general causes for drug approval time and especially its delay. Using methods such as maximum likelihood duration, one of their conclusions was that “approval times appear insignificantly related to the partisanship (...) in Congress and the White House”. The dataset was revisited by Ho and King (2007) to show how matching techniques and non-parametric methods improve the robustness of the model and causal estimates. They concluded that the average treatment effect of the Democratic Senate majority on FDA drug approval time was -35, meaning that a Democratic Senate majority delayed the drug approval on average by 35 days.

This paper uses Pearl’s causal inference framework, the Do-Calculus, and the DoWhy library to estimate the same effect, Democratic Senate majority on drug approval time. In section 1, a **model** is created and a graph is drawn that establishes the relationships between variables and makes explicit the assumptions underlying the model. In section 2, an estimand is **identified** using the backdoor approach which shows what variables need to be conditioned (or matched) on in order to have unbiased results. Section 3 the actual causal effect and average treatment effect are **estimated**. Section 4 explores methods to refute the model and general limitations. Where applicable, using the DoWhy library approach will be compared to a ‘manual’ approach.

1 - Model the graph

The grouping of variables from Carpenter (2002) and Ho (2007) was adopted.

Following the logic in Ho and King's (2007) paper, the variable groups are related as follows.

See a visualization in figure 1.

- Democratic Senate majority, clinical and epidemiological factors, disease politics and other oversight variables all affect approval times.
- Clinical and epidemiological factors also affect disease politics.
- Democratic Senate majority, disease politics and other oversight variables are affected by an unobserved variable that can be called public sentiment, or political climate.

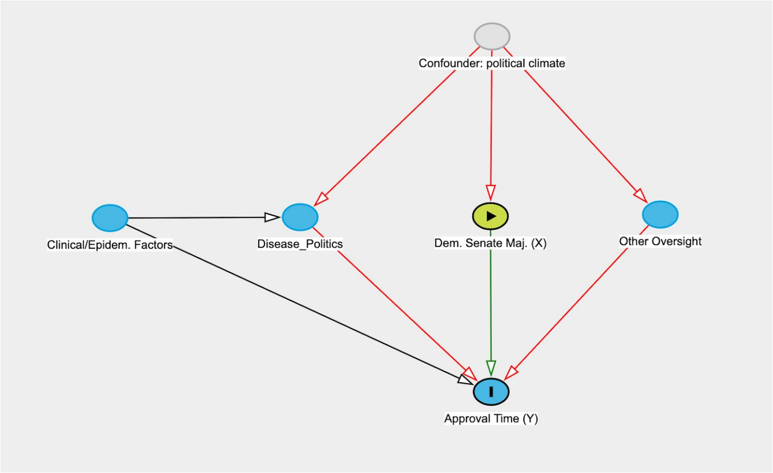


Figure 1: Causal relationships in Ho and King's (2007) paper

Model 1

Including all variables that are available, the following causal graph can be created. See figure 2.

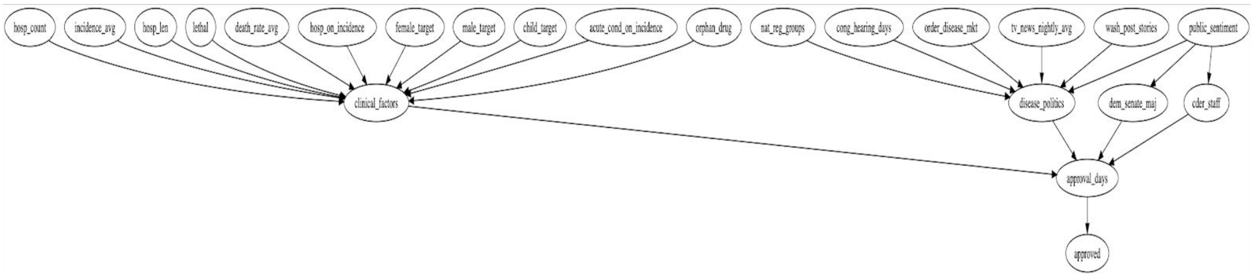


Figure 2: Complete Model

Model 2

To compare the library performance to the manual calculations, a simplified version of the graph is created. In the simplified version, child_target is the only other observable variable, in addition to democratic senate majority and approval days. See figure 3.

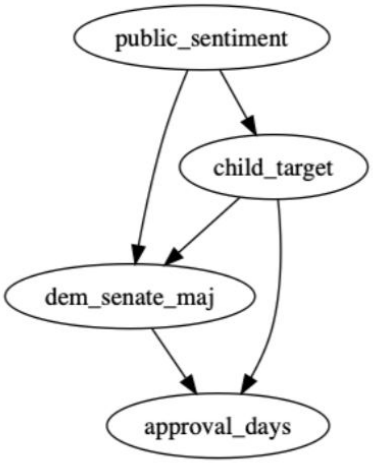


Figure 3: Simplified Model

2 - Identify the estimand

Using DoWhy

The DoWhy library uses the `identify_effect()` command to try to find estimands via the backdoor approach and using instrumental variables (iv). The backdoor approach finds variables that meet the 'backdoor criterion', which will be discussed later, and shows it is necessary to condition on them. The estimand type is ATE, the average treatment effect, which compares the difference between the average outcome (approval days) of the treatment group (Democratic Senate majority) and the control group.

Model 1

The estimand found by the library is highlighted below. Despite there being observed variables that should be confounded on (e.g. cder staff), the library does not find any variables to condition on. The library's result will be discussed further in the second model.

```
Estimand type: ate
### Estimand : 1
Estimand name: backdoor
Estimand expression:
       d
───────────────(Expectation(approval_days))
ddem_senate_maj
Estimand assumption 1, Unconfoundedness: If U→dem_senate_maj and
U→approval_days then P(approval_days|dem_senate_maj,,U) =
P(approval_days|dem_senate_maj,)
### Estimand : 2
Estimand name: iv
No such variable found!
```

Model 2

The estimand of the simplified model is highlighted below. It shows that one has to condition on child_target, in order to estimate the effect. It also highlights assumptions about external influences (U), such if there are exogenous variables (confounders) that affect Democratic Senate Majority and approval days, the probability of approval days would not change.

```
Estimand type: ate
### Estimand : 1
Estimand name: backdoor
Estimand expression:
       d
───────────────(Expectation(approval_days|child_target))
ddem_senate_maj
Estimand assumption 1, Unconfoundedness: If U→dem_senate_maj and
U→approval_days then P(approval_days|dem_senate_maj,child_target,U) =
P(approval_days|dem_senate_maj,child_target)
### Estimand : 2
Estimand name: iv
No such variable found!
```

Manual approach

Using the backdoor approach, the estimand can be found without the library as well. When a set of variables (Z) meets the backdoor criterion, the variables can be used as conditionals to isolate the direct effect of one variable (X) on another (Y). Z satisfies the backdoor criterion when no variable is a descendant of X, and the set Z blocks every indirect path between X and Y.

The graph for model 2 shows that there is a backdoor path from Democratic Senate majority to approval days via public sentiment and child target. Public sentiment is unobserved, but conditioning on child target is sufficient to block all backdoor paths. The child target variable is the only variable that meets the backdoor criterion. Conditioning on child target allows to identify the causal effect of the Democratic Senate majority on approval days.

3 - Estimate the causal effect

Using DoWhy

The DoWhy library has an `estimate_effect()` command to estimate the causal effect. The method by which the effect is estimated can be specified in the `‘method_name’` argument, and the choices are regression, stratification, matching, weighting, instrumental variable and regression discontinuity. Regression and matching will be explored further. Regression uses the conditioning set as variables and identifies the coefficient as the effect. Matching first matches observations based on a propensity score that measures similarities of the conditioning set variables, and evaluates the difference in outcomes of these variables.

Model 1

Using the matching technique, no results were found for the first method, since the library could not find common causes or confounders. Using the regression technique, a causal effect of -15.1 was estimated.

Model 2

Using the matching technique, a causal effect of -9.0 was estimated. Using the regression technique, a causal effect of -15.2 was estimated.

Manual: Model 2

Using the simplified model, the average treatment effect of a Democratic Senate Majority is -17.1. This is the difference between the average approval days of the treatment group and the average approval days of the control group when matched on child target.

Manual: Adjustment formula

How does this relate back to the adjustment formula?

In Pearl’s framework, the adjustment formula is a tool to transform an expression that includes a do() operator into one that is purely based on conditionals that can be calculated. (See appendix 2).

Adjustment formula:

$$P(Y = y|do(X = x)) = \sum_{Z} P(Y = y|X = x, Z = z)P(Z = z|X = x)$$

To apply the adjustment formula, the dataset is simplified such that approval days is a binary outcome where approval days are 'long' (Y=0) when they are above 60 days and 'short' (Y=1) when they are under or equal to 60 days. See figure 4 to see that there is a harsh decline of frequency until about 60 days.

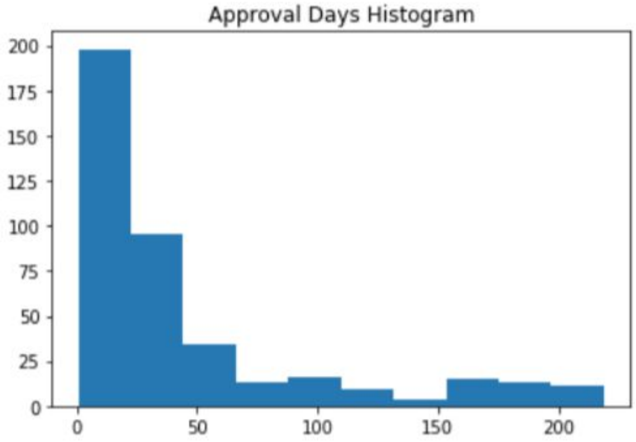


Figure 4: Approval Days Histogram

Using the adjustment formula, we want to find the average causal effect (ACE):

$$P(Y = 1|do(X = 1)) - P(Y = 1|do(X = 0)$$

which can be broken down into the probability when X = 1 and the probability when X = 0. The causal effect rule states that

$$P(Y = y|do(X = x) \;=\; \sum_{z} \frac{P(X=x, Y=y, Z=z)}{P(X=x|Z)}$$

where Z represents all parents of X that should be conditioned on.

Applied to the problem at hand, we have:

$$P(Y = 1|do(X = 0) = \sum_{z} \frac{P(X=0, Y=1,Z=z)}{P(X=0|Z)} = \frac{P(X=0, Y=1,Z=0)}{P(X=0|Z=0)} + \frac{P(X=0, Y=1,Z=1)}{P(X=0|Z=1)}$$

$$P(Y = 1|do(X = 1) = \sum_{z} \frac{P(X=1, Y=1,Z=z)}{P(X=1|Z)} = \frac{P(X=1, Y=1,Z=0)}{P(X=1|Z=0)} + \frac{P(X=1, Y=1,Z=1)}{P(X=1|Z=1)}$$

$$ACE = P(Y = 1|do(X = 1)) - P(Y = 1|do(X = 0) = 0.76 - 0.84 =- 0.08$$

The average causal effect of Democratic Senate majority on approval days is -0.08.
Given that there is a Democratic Senate majority (X=1), drugs have a probability of 0.76 to be approved quickly (Y=1). Given that there is no Democratic Senate majority (X=0), drugs have a probability of 0.84 to be approved quickly (Y=1). This analysis supports the previous approaches which favor no Democratic Senate majority for quicker approvals.

4 - Refute the model

The DoWhy library supports refutation techniques such as using placebo treatment, taking only a subset of the data and adding randomness to the model to test the robustness. For both models, the methods showed vastly different results between the actual estimate and the estimate obtained through the refutation technique, varying between -5 and -16 for the complete model and 25 and -123 for the simplified model.

Conclusion

The results are not conclusive. The DoWhy library did not compute effects based on the specified model, and the simplified result is highly sensitive to changes. The different models and different manual calculations show vastly different results which suggests either errors in the process or inconclusive data.

As concluding thoughts for the methods, depending on the situation and desired outcome, different methods may be more appropriate than others. If you are certain your model is linear, regression still can be one of the simplest straight forward approaches to obtain a causal estimate. If your model is non-linear or a parametric analysis is crucial for some other reason, matching and obtaining the difference of causal effects may be the better approach. If you are interested in a fractional change, and not the actual estimate, the Bayesian analysis could be most appropriate.

Bibliography

Carpenter, D. (2002). Groups, the media, agency waiting costs, and fda drug approval. American Journal of Political Science, 46(3), 490-505. Retrieved from https://ccl.on.worldcat.org/oclc/5545607301

Ho, D. E., Imai, K., King, G., & Stuart, E. A. (2007). Matching as Nonparametric Preprocessing for Reducing Model Dependence in Parametric Causal Inference. Political Analysis, 15(3), 199–236. doi: 10.1093/pan/mpl013. Retrieved from https://gking.harvard.edu/files/matchp.pdf

Microsoft. (n.d.). DoWhy: Making causal inference easy. Retrieved from https://microsoft.github.io/dowhy/readme.html

Appendix

Appendix 1: Variable groupings

Clinical/epidemiological variables:

- Incidence of primary indication
- Primary indication is lethal condition
- Death rate, primary indication
- Primary indication is acute condition
- Primary indication results in hospitalization
- Hospitalizations associated with indication
- Disease mainly affects women
- Disease mainly affects men
- Disease mainly affects children
- Orphan drug

Disease politics (groups and media) variables

- National and regional groups
- Nightly television news disease stories
- Washington Post disease stories
- Days of congressional hearings
- Order of disease market entry

FDA variable (Oversight)

- CDER staff

Appendix 2: Adjustment Formula

For background information, we need to understand how we can arrive from the do(x) expression to components that we can express using known conditionals. If we were to implement the intervention, X, in an experimental setting, we would be able to cut off all parents of the X variable. Even though we cannot do that in an observational setting, we can use the properties of the graph that would describe such an interventional setting to see the similarities between them.

There are two essential invariant properties we can make use of

1) $P(A = a) = P_m(A = a)$
   The probability of A is the same in both models, whether the graph was modified or not, as you can see in
2) $P(Y = y|A = a, X = x) = P_m(Y = y|A = a, X = x)$
   Removing arrows into X also does not change the probability of Y given A or X.

By combining these properties and sum over all potential values for A, we arrive at the following equation:

$$P(Y = y|do(X = x)) = \sum_A P(Y = y|X = x, A = a)P(A = a|X = x)$$

The adjustment formula says that $P(Y = y|do(X = x)) = P_m(Y = y|X = x)$, where P_m is the modified model.

Assignment 2 - DoWhy

In [97]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import dowhy
from dowhy.do_why import CausalModel
from IPython.display import Image, display
```

Data exploration

In [17]:

```
variable_dictionary = {
# Oversight
'dem_senate_maj': 'democratic senate majority',

# clinical/ epidemiological variables
'hosp_count': 'US Hospitalizations associated w/ Primary Indication Diseas
'incidence_avg': 'Incidence of primary indication',
'hosp_len': 'average length of hospitalizations (days)',
'lethal': 'Primary Indication is Lethal',
'death_rate_avg': 'Death Rate, Primary Indication (per 1000)',
'hosp_on_indicidence': 'Millions of Hospitalizations assc w/ Indication',
'female_target': 'Disease Mainly Affects Women [0, 1]',
'male_target': 'Disease Mainly Affects Men [0, 1]',
'child_target': 'Disease Mainly Affects Children [0, 1]',
'acute_cond_on_incidence': 'Primary Indication is Acute Condition [0, 1]',
'orphan_drug': 'Orphan Drug [0, 1]',

# disease politics (groups and media) variables
'nat_reg_groups': 'Number of National and Regional Groups representing Suf
'cong_hearing_days': 'Days of Cong Hrgs on Disease, 4-Year MA',
'order_disease_mkt': 'Order of Disease Market Entry for Drug i',
'tv_news_nightly_avg': 'Nightly TV News Disease Stories, 4-Year MA',
'wash_post_stories': 'Washington Post Disease Stories, 4-Year MA',

# FDA variable
'cder_staff': 'CDER staff',

# outcome
'approval_days': 'approval days',
'approved': 'approved [0, 1]'
}

df = pd.DataFrame.from_dict(variable_dictionary, orient='index')
df
```

Out[17]:

	0
dem_senate_maj	democratic senate majority
hosp_count	US Hospitalizations associated w/ Primary Indi...
incidence_avg	Incidence of primary indication

	0
hosp_len	average length of hospitalizations (days)
lethal	Primary Indication is Lethal
death_rate_avg	Death Rate, Primary Indication (per 1000)
hosp_on_indicidence	Millions of Hospitalizations assc w/ Indication
female_target	Disease Mainly Affects Women [0, 1]
male_target	Disease Mainly Affects Men [0, 1]
child_target	Disease Mainly Affects Children [0, 1]
acute_cond_on_incidence	Primary Indication is Acute Condition [0, 1]
orphan_drug	Orphan Drug [0, 1]
nat_reg_groups	Number of National and Regional Groups represe...
cong_hearing_days	Days of Cong Hrgs on Disease, 4-Year MA
order_disease_mkt	Order of Disease Market Entry for Drug i
tv_news_nightly_avg	Nightly TV News Disease Stories, 4-Year MA
wash_post_stories	Washington Post Disease Stories, 4-Year MA
cder_staff	CDER staff
approval_days	approval days
approved	approved [0, 1]

In [18]:

```
data = pd.read_csv('FDA-Carpenter_clean.csv')
data.head()
```

Out[18]:

	dem_senate_maj	hosp_count	nat_reg_groups	cder_staff	incidence_avg	hosp_len	co
0	0	75172	12	1043	115.68	3.00	
1	0	0	1	1043	100.00	0.00	
2	0	134882	9	1043	1.30	3.04	
3	0	75172	12	1043	115.68	3.00	
4	0	75172	12	1029	115.68	3.00	

1 - Model

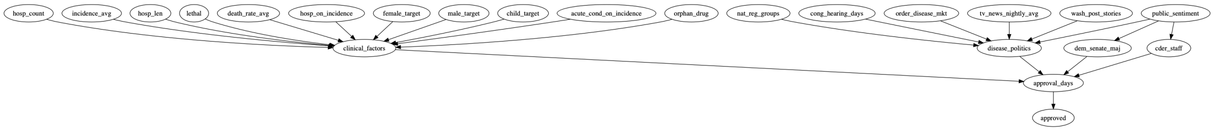
1a - Complete graph

In [184]:

```
model_a = CausalModel(
        data = data,
        treatment='dem_senate_maj',
        outcome='approval_days',
        graph="digraph {dem_senate_maj -> approval_days;\
        approval_days -> approved;\
        \
        hosp_count -> clinical_factors;\
        incidence_avg -> clinical_factors;\
        hosp_len -> clinical_factors;\
        lethal -> clinical_factors;\
        death_rate_avg -> clinical_factors;\
        hosp_on_incidence -> clinical_factors;\
        female_target -> clinical_factors;\
        male_target -> clinical_factors;\
        child_target -> clinical_factors;\
        acute_cond_on_incidence -> clinical_factors;\
        orphan_drug -> clinical_factors;\
        \
        clinical_factors -> approval_days;\
        \
        nat_reg_groups -> disease_politics;\
        cong_hearing_days -> disease_politics;\
        order_disease_mkt -> disease_politics;\
        tv_news_nightly_avg -> disease_politics;\
        wash_post_stories -> disease_politics;\
        \
        disease_politics -> approval_days;\
        \
        cder_staff -> approval_days;\
        \
        public_sentiment -> dem_senate_maj;\
        public_sentiment -> disease_politics;\
        public_sentiment -> cder_staff}"
                )
model_a.view_model()

from IPython.display import Image, display
display(Image(filename="causal_model.png"))
```

```
INFO:dowhy.do_why:Model to find the causal effect of treatment
['dem_senate_maj'] on outcome ['approval_days']
```



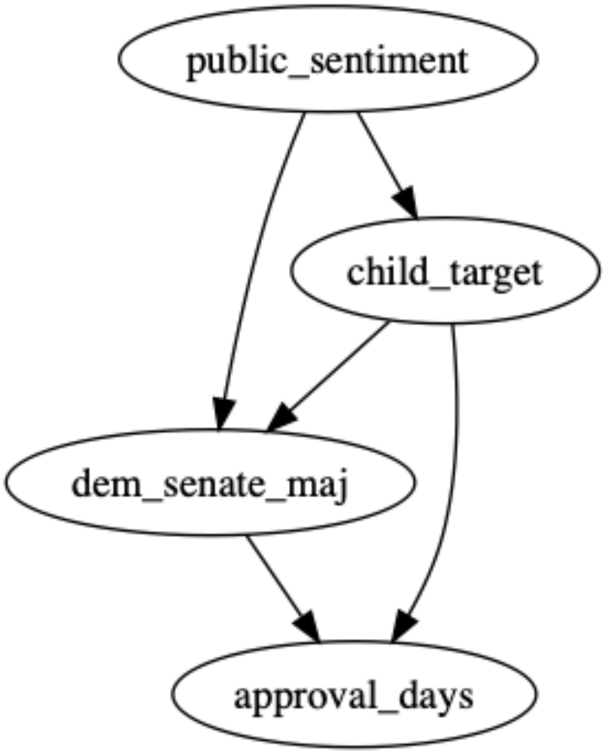
1b - Simplified model

In [185]:

```
model_b = CausalModel(
        data = data,
        treatment='dem_senate_maj',
        outcome='approval_days',
        graph="digraph {dem_senate_maj -> approval_days;\
        \
        child_target -> approval_days;\
        child_target -> dem_senate_maj;\
        \
        public_sentiment -> dem_senate_maj;\
        public_sentiment -> child_target}"
                )
model_b.view_model()

from IPython.display import Image, display
display(Image(filename="causal_model.png"))
```

```
INFO:dowhy.do_why:Model to find the causal effect of treatment
['dem_senate_maj'] on outcome ['approval_days']
```



2 - Identify

2a - Complete model

2a - Complete model

In [187]:

```
# Causal estimand
identified_estimand_a = model_a.identify_effect()
print(identified_estimand_a)
```

```
INFO:dowhy.causal_identifier:Common causes of treatment and outco
me:['public_sentiment']
WARNING:dowhy.causal_identifier:There are unobserved common cause
s. Causal effect cannot be identified.

WARN: Do you want to continue by ignoring these unobserved confou
nders? [y/n] y

INFO:dowhy.causal_identifier:Instrumental variables for treatment
and outcome:[]

Estimand type: ate
### Estimand : 1
Estimand name: backdoor
Estimand expression:
       d
───────────────(Expectation(approval_days))
ddem_senate_maj
Estimand assumption 1, Unconfoundedness: If U→dem_senate_maj and
U→approval_days then P(approval_days|dem_senate_maj,,U) = P(appro
val_days|dem_senate_maj,)
### Estimand : 2
Estimand name: iv
No such variable found!
```

2b - Simplified model

In [188]:

```python
# Causal estimand
identified_estimand_b = model_b.identify_effect()
print(identified_estimand_b)
```

```
INFO:dowhy.causal_identifier:Common causes of treatment and outco
me:['public_sentiment', 'child_target']
WARNING:dowhy.causal_identifier:There are unobserved common cause
s. Causal effect cannot be identified.

WARN: Do you want to continue by ignoring these unobserved confou
nders? [y/n] y

INFO:dowhy.causal_identifier:Instrumental variables for treatment
and outcome:[]

Estimand type: ate
### Estimand : 1
Estimand name: backdoor
Estimand expression:
       d
───────────────(Expectation(approval_days|child_target))
ddem_senate_maj
Estimand assumption 1, Unconfoundedness: If U→dem_senate_maj and
U→approval_days then P(approval_days|dem_senate_maj,child_target,
U) = P(approval_days|dem_senate_maj,child_target)
### Estimand : 2
Estimand name: iv
No such variable found!
```

1 - Estimate

1a - Complete model

Method 1: Matching

In [189]:

```
# Matching
causal_estimate_match = model_a.estimate_effect(identified_estimand_a,
        method_name="backdoor.propensity_score_matching")
print(causal_estimate_match)
print("Causal Estimate is " + str(causal_estimate_match.value))
```

```
ERROR:dowhy.causal_estimator:No common causes/confounders presen
t. Propensity score based methods are not applicable

-----------------------------------------------------------------
----------
Exception                                 Traceback (most recent
 call last)
<ipython-input-189-461dfb250191> in <module>()
      1 # Matching
      2 causal_estimate_match = model_a.estimate_effect(identifie
d_estimand_a,
----> 3         method_name="backdoor.propensity_score_matching")
      4 print(causal_estimate_match)
      5 print("Causal Estimate is " + str(causal_estimate_match.v
alue))

~/anaconda/envs/py36/lib/python3.6/site-packages/dowhy/do_why.py
 in estimate_effect(self, identified_estimand, method_name, test_
significance, method_params)
    158                     self._treatment, self._outcome,
```

Method 2: Regression

In [192]:

```
# Regression
causal_estimate_reg = model_a.estimate_effect(identified_estimand_a,
        method_name="backdoor.linear_regression",
        test_significance=True)
print(causal_estimate_reg)
print("Causal Estimate is " + str(causal_estimate_reg.value))
```

```
INFO:dowhy.causal_estimator:INFO: Using Linear Regression Estimat
or
INFO:dowhy.causal_estimator:b: approval_days~dem_senate_maj

*** Causal Estimate ***

## Target estimand
Estimand type: ate
### Estimand : 1
Estimand name: backdoor
Estimand expression:
       d
───────────────(Expectation(approval_days))
ddem_senate_maj
Estimand assumption 1, Unconfoundedness: If U→dem_senate_maj and
U→approval_days then P(approval_days|dem_senate_maj,,U) = P(appro
val_days|dem_senate_maj,)
### Estimand : 2
Estimand name: iv
No such variable found!
```

3b - Simplified model

Method 1: Matching

In [179]:

```python
# Matching
causal_estimate_match = model_b.estimate_effect(identified_estimand_b,
        method_name="backdoor.propensity_score_matching")
print(causal_estimate_match)
print("Causal Estimate is " + str(causal_estimate_match.value))
```

```
INFO:dowhy.causal_estimator:INFO: Using Propensity Score Matching
Estimator
INFO:dowhy.causal_estimator:b: approval_days~dem_senate_maj+child
_target

*** Causal Estimate ***

## Target estimand
Estimand type: ate
### Estimand : 1
Estimand name: backdoor
Estimand expression:
       d
───────────────(Expectation(approval_days|child_target))
ddem_senate_maj
Estimand assumption 1, Unconfoundedness: If U→dem_senate_maj and
U→approval_days then P(approval_days|dem_senate_maj,child_target,
U) = P(approval_days|dem_senate_maj,child_target)
### Estimand : 2
Estimand name: iv
```

Method 2: Regression

In [191]:

```python
# Regression
causal_estimate_reg = model_b.estimate_effect(identified_estimand_b,
        method_name="backdoor.linear_regression",
        test_significance=True)
print(causal_estimate_reg)
print("Causal Estimate is " + str(causal_estimate_reg.value))
```

```
INFO:dowhy.causal_estimator:INFO: Using Linear Regression Estimat
or
INFO:dowhy.causal_estimator:b: approval_days~dem_senate_maj+child
_target

*** Causal Estimate ***

## Target estimand
Estimand type: ate
### Estimand : 1
Estimand name: backdoor
Estimand expression:
       d
───────────────(Expectation(approval_days|child_target))
ddem_senate_maj
Estimand assumption 1, Unconfoundedness: If U→dem_senate_maj and
U→approval_days then P(approval_days|dem_senate_maj,child_target,
U) = P(approval_days|dem_senate_maj,child_target)
### Estimand : 2
Estimand name: iv
No such variable found!

## Realized estimand
b: approval_days~dem_senate_maj+child_target
## Estimate
Value: -15.189955215386018

## Statistical Significance
p-value: 0.002

Causal Estimate is -15.189955215386018
```

3 - Manual approach

ATT via matching

In [202]:

```
# Match on child target and compute treatment effect

data = pd.read_csv('FDA-Carpenter_clean.csv')
simplified_ATT = data[['dem_senate_maj', 'approval_days', 'child_target']]

# control
control_Zis0 = simplified_ATT[(simplified_ATT['dem_senate_maj'] == 0)
                              & (simplified_ATT['child_target'] == 0)]
control_Zis1 = simplified_ATT[(simplified_ATT['dem_senate_maj'] == 0)
                              & (simplified_ATT['child_target'] == 1)]

# treatment
tmt_Zis0 = simplified_ATT[(simplified_ATT['dem_senate_maj'] == 1)
                          & (simplified_ATT['child_target'] == 0)]
tmt_Zis1 = simplified_ATT[(simplified_ATT['dem_senate_maj'] == 1)
                          & (simplified_ATT['child_target'] == 1)]


Zis0_effect = (np.mean(tmt_Zis0['approval_days']) -
               np.mean(control_Zis0['approval_days']))
Zis1_effect = (np.mean(tmt_Zis1['approval_days']) -
               np.mean(control_Zis1['approval_days']))
ATT = Zis0_effect - Zis1_effect
ATT
```

Out[202]:

-17.0599720553041

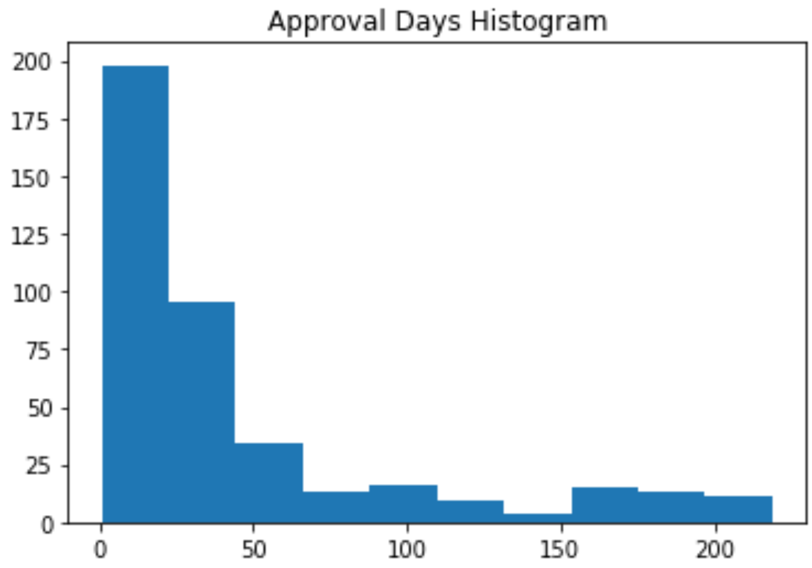
Adjustment Formula - Manual Bayesian Analysis

In [205]:

```
plt.hist(data.approval_days)
plt.title('Approval Days Histogram')
plt.plot()
```

Out[205]:

```
[]
```



In [206]:

```
# Edited table
data = pd.read_csv('FDA-Carpenter_clean.csv')
simplified = data[['dem_senate_maj', 'approval_days', 'child_target']].cop
simplified['approval_days'] = simplified['approval_days'].apply(lambda x :
simplified.head()
```

Out[206]:

	dem_senate_maj	approval_days	child_target
0	0	1	0
1	0	0	0
2	0	1	0
3	0	1	0
4	0	1	0

In [207]:

```
def get_joint_3(df, x_value, y_value, z_value,
                          X = 'dem_senate_maj', Y = 'approval_days', Z = '
    joint_count = len(df[(df[X] == x_value) & (df[Y] == y_value) & (df[Z]
    total = len(df)
    return joint_count/ total


def get_joint_2(df, x_value, z_value,
                          X = 'dem_senate_maj', Z = 'child_target'):
    joint_count = len(df[(df[X] == x_value) & (df[Z] == z_value)])
    total = len(df)
    return joint_count/ total


def get_marginal(df, z_value, Z = 'child_target'):
    marginal_count = len(df[(df[Z] == z_value)])
    total = len(df)
    return marginal_count/ total


def get_conditional(df, x_value, z_value,
                                X = 'dem_senate_maj', Z = 'child_target'):
    joint_xz = get_joint_2(df, x_value, z_value)
    marginal_z = get_marginal(df, z_value)
    return joint_xz/ marginal_z

```

In [216]:

```
dem = (get_joint_3(simplified, 1, 1, 1)/ get_conditional(simplified, 1, 1)
          get_joint_3(simplified, 1, 1, 0)/ get_conditional(simplified, 1,
not_dem = (get_joint_3(simplified, 0, 1, 1)/ get_conditional(simplified, 0
          get_joint_3(simplified, 0, 1, 0)/ get_conditional(simplified, 0,

ACE = not_dem - dem

print(f'P(Y=1|do(X=0) = {not_dem})')
print(f'P(Y=1|do(X=1) = {dem})')
print(f'ACE = {not_dem - dem}')
```

```
P(Y=1|do(X=0) = 0.7592315645655625)
P(Y=1|do(X=1) = 0.841539179774474)
ACE = -0.0823076152089115
```

4 - Refute

4a - Complete Model

In [79]:

```
# Replacing treatment with a random (placebo) variable
res_placebo=model.refute_estimate(identified_estimand_a, estimate,
        method_name="placebo_treatment_refuter", placebo_type="permute")
print(res_placebo)
```

```
INFO:dowhy.causal_estimator:INFO: Using Propensity Score Matching
Estimator
INFO:dowhy.causal_estimator:b: approval_days~placebo+wash_post_st
ories

Refute: Use a Placebo Treatment
Estimated effect:(-14.15655634110429,)
New effect:(-5.136427060122696,)


/Users/annapauxberger/anaconda/envs/py36/lib/python3.6/site-packa
ges/dowhy/causal_estimators/propensity_score_matching_estimator.p
y:51: FutureWarning: `item` has been deprecated and will be remov
ed in a future version
  control_outcome = control.iloc[indices[i]][self._outcome_name].
item()
```

In [80]:

```python
# Removing a random subset of the data
res_subset=model.refute_estimate(identified_estimand_a, estimate,
        method_name="data_subset_refuter", subset_fraction=0.9)
print(res_subset)

```

```
INFO:dowhy.causal_estimator:INFO: Using Propensity Score Matching
Estimator
INFO:dowhy.causal_estimator:b: approval_days~dem_senate_maj+wash_
post_stories

Refute: Use a subset of data
Estimated effect:(-14.15655634110429,)
New effect:(-16.301956866666657,)


/Users/annapauxberger/anaconda/envs/py36/lib/python3.6/site-packa
ges/dowhy/causal_estimators/propensity_score_matching_estimator.p
y:51: FutureWarning: `item` has been deprecated and will be remov
ed in a future version
  control_outcome = control.iloc[indices[i]][self._outcome_name].
item()
```

In [81]:

```
# Add random seed

res_subset=model.refute_estimate(identified_estimand_a, estimate,
        method_name="data_subset_refuter", subset_fraction=0.9, random_see
print(res_subset)
```

```
INFO:dowhy.causal_estimator:INFO: Using Propensity Score Matching
Estimator
INFO:dowhy.causal_estimator:b: approval_days~dem_senate_maj+wash_
post_stories

Refute: Use a subset of data
Estimated effect:(-14.15655634110429,)
New effect:(-6.816135866206897,)


/Users/annapauxberger/anaconda/envs/py36/lib/python3.6/site-packa
ges/dowhy/causal_estimators/propensity_score_matching_estimator.p
y:51: FutureWarning: `item` has been deprecated and will be remov
ed in a future version
  control_outcome = control.iloc[indices[i]][self._outcome_name].
item()
```

4a - Simplified Model

In [217]:

```
# Replacing treatment with a random (placebo) variable
res_placebo=model.refute_estimate(identified_estimand_b, estimate,
        method_name="placebo_treatment_refuter", placebo_type="permute")
print(res_placebo)
```

```
INFO:dowhy.causal_estimator:INFO: Using Propensity Score Matching
Estimator
INFO:dowhy.causal_estimator:b: approval_days~placebo+child_target

Refute: Use a Placebo Treatment
Estimated effect:(-14.15655634110429,)
New effect:(25.106917635582814,)


/Users/annapauxberger/anaconda/envs/py36/lib/python3.6/site-packa
ges/dowhy/causal_estimators/propensity_score_matching_estimator.p
y:51: FutureWarning: `item` has been deprecated and will be remov
ed in a future version
  control_outcome = control.iloc[indices[i]][self._outcome_name].
item()
```

In [218]:

```
# Removing a random subset of the data
res_subset=model.refute_estimate(identified_estimand_b, estimate,
        method_name="data_subset_refuter", subset_fraction=0.9)
print(res_subset)

```

```
INFO:dowhy.causal_estimator:INFO: Using Propensity Score Matching
Estimator
INFO:dowhy.causal_estimator:b: approval_days~dem_senate_maj+child
_target
/Users/annapauxberger/anaconda/envs/py36/lib/python3.6/site-packa
ges/dowhy/causal_estimators/propensity_score_matching_estimator.p
y:51: FutureWarning: `item` has been deprecated and will be remov
ed in a future version
  control_outcome = control.iloc[indices[i]][self._outcome_name].
item()

Refute: Use a subset of data
Estimated effect:(-14.15655634110429,)
New effect:(-123.43703207210892,)
```

In [219]:

```
# Add random seed

res_subset=model.refute_estimate(identified_estimand_b, estimate,
        method_name="data_subset_refuter", subset_fraction=0.9, random_see
print(res_subset)
```

```
INFO:dowhy.causal_estimator:INFO: Using Propensity Score Matching
Estimator
INFO:dowhy.causal_estimator:b: approval_days~dem_senate_maj+child
_target

Refute: Use a subset of data
Estimated effect:(-14.15655634110429,)
New effect:(17.853817707382547,)


/Users/annapauxberger/anaconda/envs/py36/lib/python3.6/site-packa
ges/dowhy/causal_estimators/propensity_score_matching_estimator.p
y:51: FutureWarning: `item` has been deprecated and will be remov
ed in a future version
  control_outcome = control.iloc[indices[i]][self._outcome_name].
item()
```